

Robotics Research Technical Report

generatorium omnis laboris ex machina

B I R D
A high-level interpreter for
PLOTING CURVES IN THREE DIMENSIONS
Using NCAR

by

Alex Cimino

Technical Report No.166
Robotics Report No.47

June, 1985

TO COME IN 100 C.1
Cimino, Alex

BIRD

New York University
Courant Institute of Mathematical Sciences

Computer Science Division
251 Mercer Street New York, N.Y. 10012





B I R D
A high-level interpreter for
PLOTING CURVES IN THREE DIMENSIONS
Using NCAR

by

Alex Cimino

Technical Report No.166
Robotics Report No.47

June, 1985

New York University
Department of Computer Science
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, N.Y. 10012

Work on this paper has been supported in part by Office of Naval Research Grant N00014-82-K-0381, and by grants from the Digital Equipment Corporation, the IBM Corporation, and by National Science Foundation CER Grant No. DCR-83-20085.

NEW YORK UNIVERSITY
COLLEGE OF ENGINEERING
251 Mercer St. New York, N.Y. 10012

BIRD

A high-level interpreter for
PLOTING CURVES IN THREE DIMENSIONS
Using NCAR

by

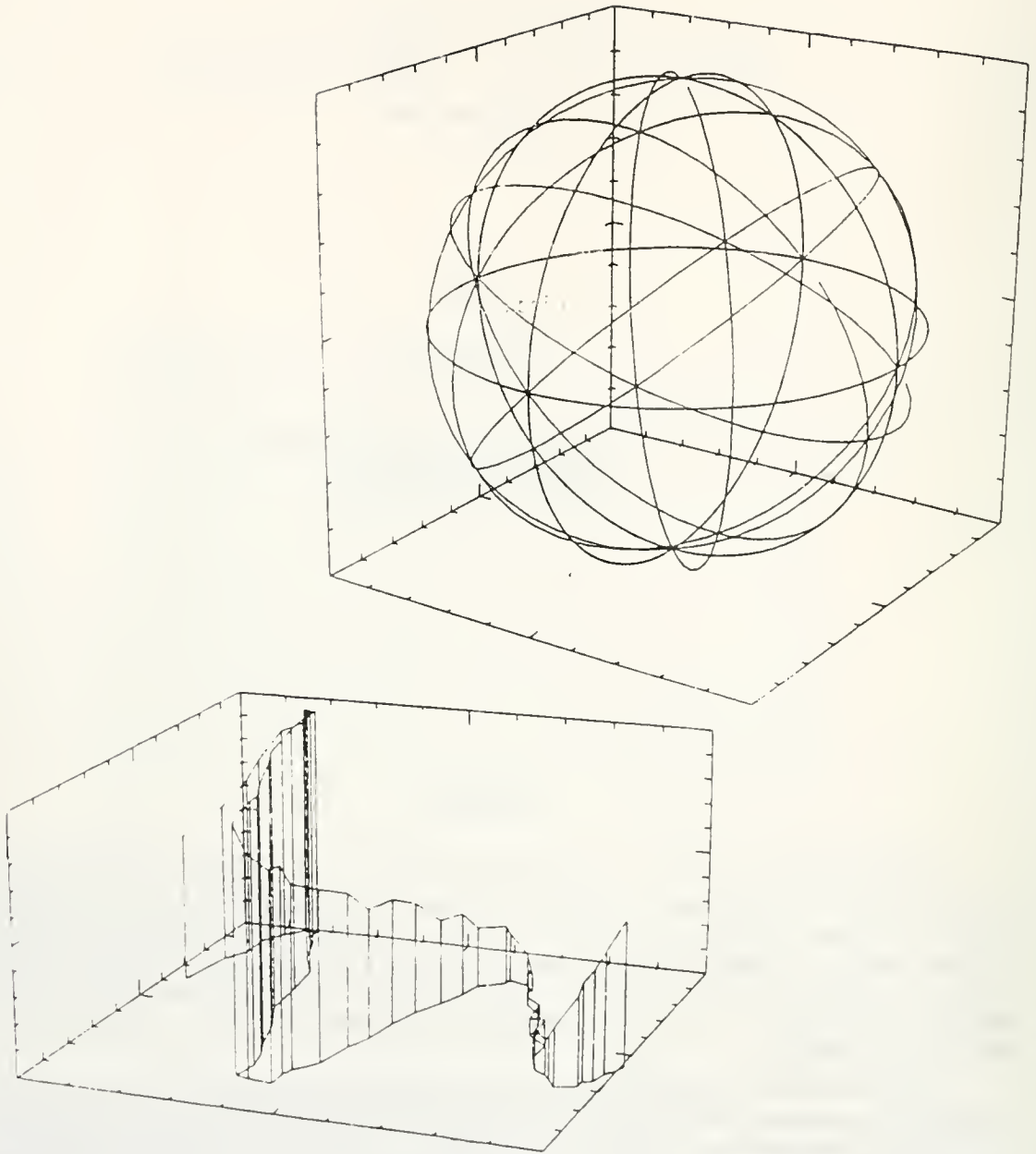
Alex Cimino

Courant Institute of Mathematical Sciences
New York University
New York, N.Y. 10012

ABSTRACT

NCAR (National Center for Atmospheric Research) is a FORTRAN graphics package that allows a fairly large number of options for, among many other things, plotting curves in three dimensions. However, the direct calling of NCAR routines from a FORTRAN program results in a considerable amount of details to take care of. This is both time-consuming and error-prone. To free the user from almost all low-level considerations, I have prepared a FORTRAN 77 interpreter called BIRD. (Doesn't a 3-D curve make you think of the flight of a bird?)

BIRD reads a file of high-level instructions and then calls NCAR subroutines with the appropriate parameters, thereby preparing an output file ready to be plotted. Furthermore, it identifies most user mistakes that, if undetected, would cause a crash; appropriate error messages are then printed out and the execution is stopped. This paper assumes that BIRD is run on the VMS operating system.



Example of 3-D curve displays. For more, see appendix C.

INTRODUCTION

Throughout this paper, the word “curve” will indicate a collection of consecutive segments. Obviously, the greater the number of points used to approximate a given curve, the better the quality of the plot; however, there is a limit to the number of those points, namely 10000.

For each three dimensional curve to be displayed, a file of x, y, and z-coordinates is needed. The first line of the file must include the number of points the curve is made up of. Such number should clearly be less or equal to the number of 3-D coordinates included in your file. (If it is less, the remaining coordinates are ignored.) The following format should be used :

```
<Number of points>
x1 y1 z1
x2 y2 z2
....
xN yN zN
```

Notice that each individual line does not require any particular formatting.

A file which includes all BIRD commands, as will be explained later on, is then needed.

The name of this file *must* be :

bird.

At this point, after your files of commands is ready, if you have the executable file *bird.exe* , you can simply type

```
run bird
```

While the execution goes on, the commands are echoed on the standard output as they are read in, some useful information is printed out, and an output file named

ncarmeta.fil

is created. (Appendix B explains how to display such a file.) If you have the original source file *bird.for*, you must first compile it in the following way :

```
fortran bird
link bird, 'ncarlib'
```

In case no command file named "bird." is present when attempting to run BIRD, an error message will be issued.

BASIC B I R D COMMANDS

The command file "bird." must have the following format :

```
<Group of commands and parameters>
.
<Optional other group>
.
....
<Optional other group>
.
```

All commands and names (in other words, everything that is read by the FORTRAN program as a string) *must* be left-justified. Numbers need not be formatted in any particular way. Notice the use of the period (.) at the beginning of a line to indicate the end of one plotting and the beginning of another.

In order to display a curve in three dimensions, two things are essential : a file of coordinates (as described in the introduction) and the point in 3-space, called *eye*, from which we view at the curve. The eye coordinates are assumed to be expressed in the same unit of measurement as the points on the curve. To read in a file of coordinates, the following command should be used :

CUR_{*n*}
<filename>

Filename must be a string of no more than 50 characters (including directory specification, if necessary.) The parameter *n* is a number between 1 and 9 (if 1, it can be omitted) that represents the register to which the curve is assigned. In this way, up to 9 curves can be stored and later referred to by their number. If two or more curves are labeled with the same number, only the last one will be available.

By default, all curves read in — except those that have been discarded because a new curve has been assigned the same number — are shown in every plot. If only a subset of all curves stored is to be displayed, then the following command should be used :

SHOW
<Curvenumber>
<Optional other curve#>
....
<Optional other curve#>

Curve numbers must be integers between 1 and 9. If no curve numbers follow the SHOW command, or if a number that hasn't yet been assigned to any curve appears, an error message will appear and the program will stop. The SHOW command is in effect until superseded by another SHOW. However, if a new curve is read in, it will replace the old one (if its number was among the arguments of the last SHOW) or it will be automatically added to the list of the curves to include in each plot.

The way to specify the EYE position is the following :

EYE
x-coord. y-coord. z-coord.

The position of the EYE *must* lie outside of the parallelepiped circumscribing the curve (or the user-defined window — see next section); otherwise, an error message will be produced

and execution will halt. An abnormal termination can also be caused by neglecting to specify the EYE coordinates for the first plot. Whenever a new EYE command is issued, the old eye coordinates are replaced by the new ones, which will be used from then on. If no EYE command is present in a plot other than the first, the latest EYE coordinates will be assumed.

FRAMES OF REFERENCE AND WINDOWS

A lonely curve viewed in an empty space may not be of much help; a frame of reference is very useful to visualize it. In BIRD, a frame of reference is a set of up to six coordinate planes surrounding the given curve. The term "coordinate plane" indicates any plane parallel to either the xy -, the yz - or the xz - plane. Depending on which curves are going to be displayed together in the same given plot, BIRD will automatically define the 3-D *minimal window*, i.e. the parallelepiped just large enough to contain all those curves inside. Because of such default action, size differences between curves cannot be appreciated unless the curves are shown together. However, the user has the option of defining the window that will actually be used in the plot. To do so, give the command

```
WIN
<left   right>
<front  back>
<bottom top>
```

The new window will now be a parallelepiped whose lower left-hand side corner has coordinates (left, front, bottom) and whose upper right-hand side corner is (right, back, top). Such choice of names is quite natural because BIRD uses a dextrous coordinate system. (See fig. 1) It should be kept in mind, though, that those names refer to the coordinate system,

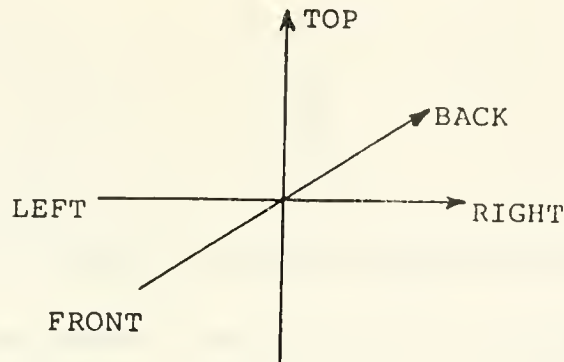


Figure 1.

and not to what we actually see; so, e.g., if we move the eye position "behind" the curve, then the left plane will actually appear to the right of the plot. This reflects the fact that "left" is the plane perpendicular to the x -axis whose x -coordinate is the smallest of all x -coordinates of the points on the curve.

User-defined windows must be at least as large as the minimal ones for the given set of curves; BIRD will not clip curves to fit a window that is too small. The default minimal window can be restored with the command

DWIN

To specify a frame of reference, the following command has to be entered :

FRA

followed by any combination (one, several or none) of the following mnemonics, each on a separate line :

le
ri
fr
ba
bo
to

They stand for, respectively, left, right, front, back, bottom, top.

If no FRA command appears in the first plot then, by default, the following planes are assumed : bo, le, ba. Whenever a new FRA command appears, the old set of planes is replaced with the new one, which will be used from then on.

PROJECTIONS

The curve assigned to number 1 can be projected on any of the planes of reference. If this is desired, the command :

PRO

should be used, followed by any combination of the same group of mnemonics seen for FRA. The same defaults as in FRA apply, with the exception that if no PRO command appears in the first plot then no projections are assumed.

Another option that can be used to enhance the sense of three dimensionality for the curve number 1 is the so-called *fencing*. If the request

FEN n

is included, followed by any combination of the mnemonics seen above, a set of segments going perpendicularly from points on the curve to each specified plane of reference will be drawn. The parameter n should be an integer between 1 and 9 (if 1, it may be omitted.) It

specifies the desired density of the fence; a projection segment will be drawn for every n points on the curve. This means that 1 is the densest and 9 is the least dense. It should be remembered that, no matter what density is used, more than two or three fences are very likely to make the plot hopelessly messy. Exactly the same defaults as in PRO apply.

CAPTION: and STOP commands

Each plot can include a caption of up to 41 characters. To specify a caption, use the command :

CAP
<Here you enter your caption>

Lower case letters should *not* be included in the caption because NCAR cannot print them. If no caption is specified for a given plot, then no caption is assumed. If the command CAP appears more than once in the same plot, then only the last caption will appear.

The command

STOP

can be entered at any point in the command file. It causes execution to be halted. All the plots preceding the one containing the STOP will be included in the output file; all the others will be ignored. If not all plots are to be displayed, the command STOP makes it unnecessary to delete a part of the file "bird." (Which may be needed in the future.)

SEMI-INTERACTIVE MODE

Regretfully, NCAR cannot be used interactively, and therefore neither can BIRD. Nevertheless, BIRD commands may be entered interactively — rather than being read from a file — if so is desired. To enter the INTERactive mode, include the command

INT

in your "bird." file; from that point on, BIRD will stop and wait for you to enter your commands interactively. No prompts will appear, but BIRD will continue to echo your requests and to print some other information, as it does ordinarily. To switch back to the Non-INTERactive mode, enter the request

NINT

New commands will be read once again from the file "bird," starting from the line following the INT request.

AN ILLUSTRATIVE EXAMPLE

Let "sphere3." be the file of points generated by the following FORTRAN program

```

      OPEN (FILE = 'sphere3.', UNIT = 12, STATUS = 'NEW',
+       ACCESS = 'SEQUENTIAL', FORM = 'FORMATTED')

      D = 2 * 3.141597 / 50
      F = SQRT(2.)
      WRITE(12, *), 200

      DO 10 I = 0, 49
        T = D * FLOAT(I)
        WRITE(12, *), .0, SIN(T), COS(T)
10      CONTINUE

      DO 20 I = 0, 49
        T = D * FLOAT(I)
        WRITE(12, *), SIN(T), 0., COS(T)
20      CONTINUE

      DO 30 I = 0, 49
        T = D * FLOAT(I)
        WRITE(12, *), SIN(T)/F, SIN(T)/F, COS(T)
30      CONTINUE

      DO 40 I = 0, 49
        T = D * FLOAT(I)
        WRITE(12, *), -SIN(T)/F, SIN(T)/F, COS(T)
40      CONTINUE

      CLOSE (UNIT = 12)

      END

```

Then, the following BIRD program will produce the plot shown in Fig.2 :

```

CUR
sphere3.
EYE
4.3 -7.3 4.1
WIN
-3. 3.
-3. 3.
-3. 3.
PRO
bo
le
ba
.

```

Notice that each of the sides of the 3-D window we have chosen is about three times larger than the corresponding side in the "minimal" window, i.e. the one just large enough to

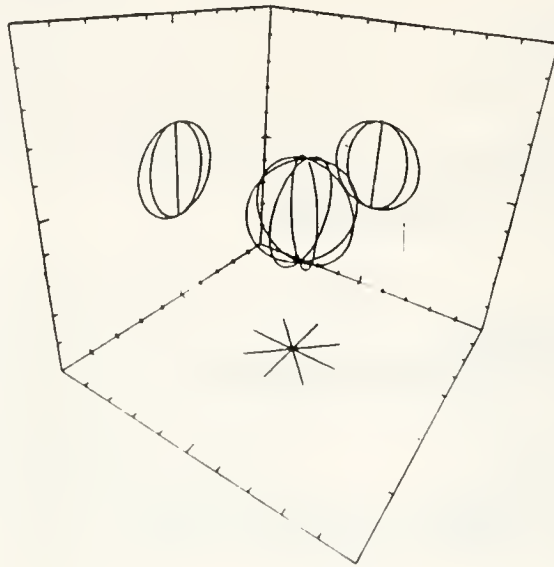


Figure 2.

contain the curve.

Appendix A

LIST OF ALL BIRD COMMANDS

An asterisk (*) indicates that the command is of the form `COMMANDn`, where *n* is an integer between 1 and 9. (May be omitted if 1.) A dagger (†) indicates that *no* arguments are expected in the line(s) immediately after.

CAP

CUR*

EYE

DWIN †

FEN*

FRA

INT †

NINT †

PRO

SHOW

STOP †

WIN

Appendix B

HOW TO DISPLAY A NCAR OUTPUT FILE

After you receive the prompt (\$), give the command :

`ncar`

Notice that, at New York University, you cannot invoke NCAR unless after logging in you give the command

`@nyu$ncar:ncar`

(If you plan to use NCAR several times, you might as well include that command in your login file.) At this point you will be asked to enter a number of options. The new prompt is :

META OPTION :

The four basic options are :

- i) `read <Optional filename>`
- ii) `device tt <Here you must specify the device type>`
- iii) `plot`
- iv) `exit`

"read" (abbreviated `re`) takes by default the latest version of a file called `NCARMETA.FIL` containing the graph to be plotted. "device `tt`" (abbr. `de tt`) allows the user to specify the graphic device that is going to be used. If you are using a terminal with graphic capabilities you should indicate your terminal type (e.g., `VT100`, or `ADM3`); if you want a hard copy, the device type you should specify is : `ZETA` .

After you have specified both the file to be read and the device type, you can give the "plot" (abbr. pl) command, that will display your plot on the terminal or send it to the plotter. If your output file contains more than one graph, just give the "plot" command as many times as necessary to display all pictures. The option "exit" (abbr. ex) allows the user to leave NCAR.

It is possible to switch from one output device to another at any time. For example, entering the following group of commands

```
read
de tt vt100
pl
pl
de tt zeta
pl
de tt vt100
pl
```

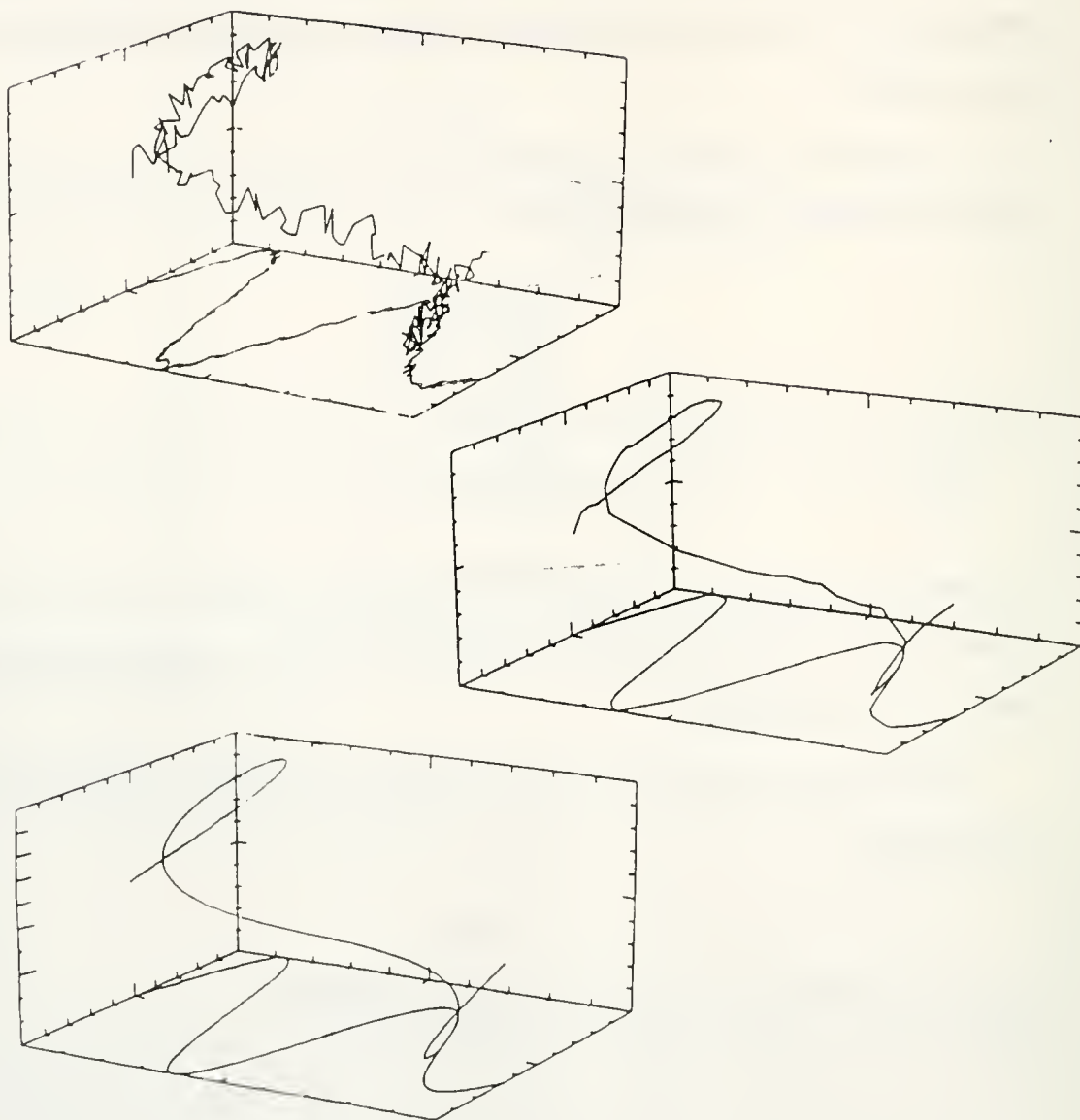
will cause the first two plots to be displayed on the screen, the third to be sent to the plotter, and the fourth to appear on the screen once again. (We are here assuming that the output file contains at least four graphs.)

In order to view a given plot, it is *not* necessary to go through all the previous ones. The instruction

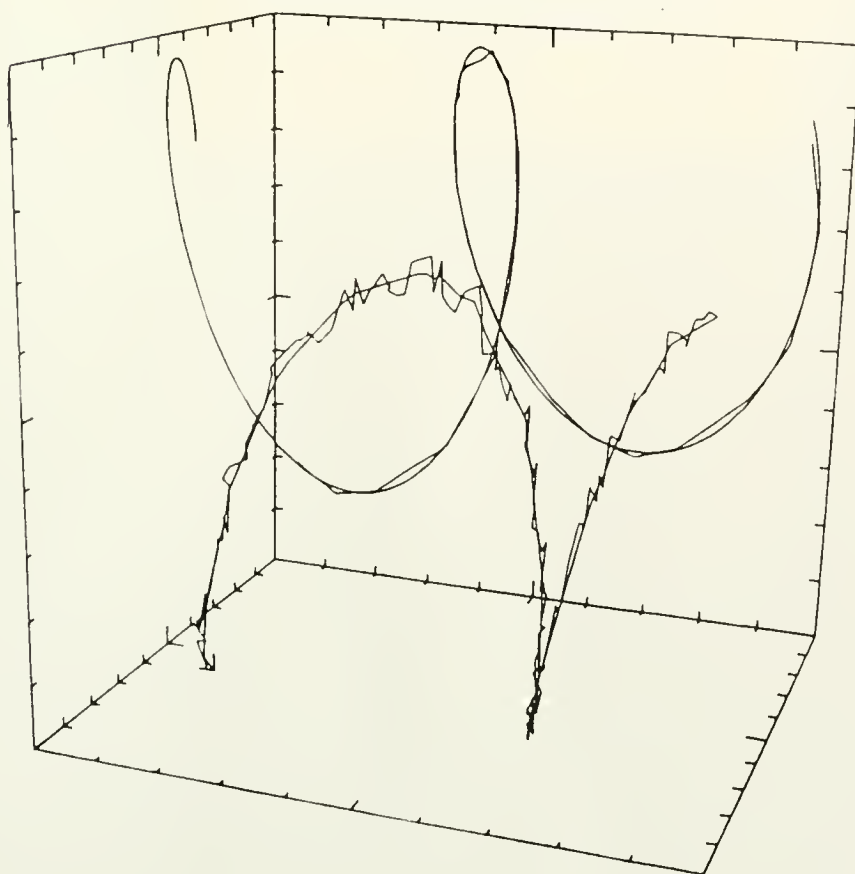
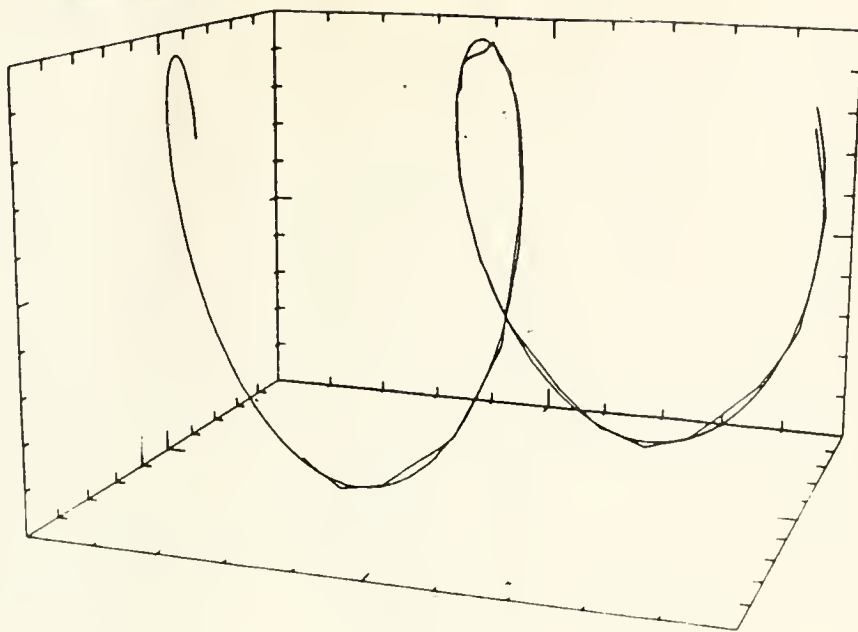
```
jump  $n$ 
```

can be used, causing the n -th graph (if present) to be displayed.

Appendix C



While smoothing a curve, one can have a feeling for what is going on.



Superposition of Curves.

This book may be kept

FOURTEEN DAYS

A fine will be charged for each day the book is kept overtime.

[illegible]

NYU COMPSCI TR-166
Cimino, Alex c.1

BIRD

NYU COMPSCI TR-166
Cimino, Alex c.1

BIRD

DATE DUE	BORROWER'S NAME

LIBRARY
N.Y.U. Courant Institute of
Mathematical Sciences
251 Mercer St.
New York, N. Y. 10012

